Title
Teammatic: A Mixed Initiative Interface for Team Composition with Multiple Constraints

Author
Thio, Carolyn

Publication Date
2017

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA, SAN DIEGO

Teammatic: A Mixed Initiative Interface for Team Composition with Multiple Constraints

A Thesis submitted in partial satisfaction of the
requirements for the degree
Master of Science

in

Computer Science

by

Carolyn Thio

Committee in charge:

Steven Dow, Chair
Scott Klemmer
Nadir Weibel

2017

The Thesis of Carolyn Thio is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California, San Diego

2017

DEDICATION

To my family for the constant love and support.

Thanks for always believing in me no matter what.

We did it!

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

I am so grateful to have had the opportunity to embark on this journey. To begin, I would like to thank Steven Dow for constantly providing support and guidance throughout the course of the project. I have learned so much from you, and your passion for this project has kept me motivated every step of the way.

Thank you to both Scott Klemmer and Nadir Weibel for providing feedback and advice on how to improve my thesis.

I would also like to thank my team that has been with me through this entire process. Thanks to Wentao Guan, Yikun Huang, Jace Lu, and Alex Tran. I would not have been able to complete this thesis without all your help!

ABSTRACT OF THE THESIS

Teammatic: A Mixed Initiative Interface for Team Composition with Multiple Constraints

by

Carolyn Thio

Master of Science in Computer Science

University of California San Diego, 2017

Steven Dow, Chair

Project teams are increasingly common in classes. However, simultaneously assembling dozens of effective teams is challenging because it is difficult to take into consideration several different constraints. This thesis develops an interface that facilitates the process of forming teams within classes. It introduces

Teammatic, a mixed initiative interface for instructors to work in conjunction with an algorithm. We created an algorithm that will account for several constraints, as inputted by the instructor, in order to create teams among the students. Common constraints include distributing roles and ensuring overlapping schedules amongst teams. Our results show that Teammatic produces teams that have equal or higher compatibility scores than teams created manually. Furthermore, Teammatic allows instructors to form teams given various constraints in significantly less time.

Chapter 1

Introduction

A team is a group of individuals working together to complete a common task or goal. As teams have become increasingly standard in industry and education, there is much interest in what makes teams most effective. Each individual holds their own unique set of characteristics and with the right combination of individuals, a team can work more effectively together.

Team-based courses help students build these skills. Previous research has shown that students learn proper teamwork skills by being placed on effective teams [23]. It is simple to create random teams, but randomly placing students together does not consistently produce teams that are effective in achieving their goal. This thesis introduces Teammatic, the first mixed initiative user interface that allows professors to create teams with specific constraints in mind. We see the need for students to benefit from collaborating in an effective team; however, we realize that each individual has unique attributes that must be accounted, such as their leadership qualities or personal schedules. With Teammatic, instructors are able to input constraints, which will output suggested team formations, while still having the freedom to move students as they see fit. Thus, with Teammatic, creating teams takes significantly less time, produces higher compatibility scores, and increases confidence from the instructor creating the teams [Section 5.3].

1.1 Problem Setting

With the increased demand of teamwork required in industry, group projects allow students to learn key skills, such as teamwork, problem-solving, communication, and leadership [12]. There are several different educational benefits students can gain from working with one another in a team. Collaboration allows students to gain a better understanding of the subject material, as well as encourage cognitive growth [9]. Gaining proper teamwork skills involves proper student interaction with other team members.

However, research has been shown that placing students in ineffective teams, characterized by conflict, unclear goals, or mismanagement, creates problems that are not beneficial to developing skills for the students involved [12]. Teammatic attempts to ease the process of forming effective teams, specifically within an educational setting.

1.2 Difficulty of the Problem

Team formation can be difficult given the different schedules and traits of various individuals. In the educational setting, students have other commitments and priorities that go beyond the classroom. Given these varying schedules, it may be hard for members of the team to communicate with one another, if the team cannot all find a time to meet. In addition, each individual has unique characteristics that define their work habits and social traits. Studies have shown certain characteristics are important in producing an effective team, and in order to create these teams, it is important to take each of these traits into consideration [2]. There are several attributes that are important in determining how individuals should be placed together, such as: communication, team size, and proper leadership [25].

Chapter 2

Related Work

2.1 Previous Research on Team Analysis

Past research has shown that certain characteristics among the members of a team have an effect on the overall team performance. Individuals gain proper teamwork skills when they are placed in an effective team rather than one filled with conflicts and disagreements [13]. The varying characteristics of an individual, such as their leadership qualities and technical skills, influence the team's effectiveness indirectly through the nature of interdependent activity among team members [19]. We are interested in the performance of the team, given its specific composition – the unique combination of individuals [5].

Each member of a team has the potential to not only influence the overall team, but also each of the other team members [19]. While teams are affected by its members, past research as shown that there are other factors that can affect the team's performance, including the complexity of the task at hand and overall group cohesion [11,27]. Studies have shown how conflict among team members is proven to be disruptive within a team [10]. This stresses the importance of creating effective teams for students to learn and develop essential skills.

2.1.1 Leadership

Leadership within a team has the potential to both positively influence the team learning and knowledge applications within a group [30]. Team leaders are essential within a group setting since they play a pivotal role in both maintaining the environment and coordinating tasks within a team [22]. An effective leader on a team can provide guidance to other members, maintain organization, and foster collaboration with one another.

We provide a constraint to allow instructors to distribute the leaders within the classroom across teams during the formation process. Students can indicate whether they prefer to lead, prefer not to lead, or can play either role. Utilizing this data, instructors have the ability to distribute the leaders across teams.

2.1.2 Demographics

Inclusion is important within the educational setting. For individuals belonging to a minority group, there is a high risk that these students may drop out of the given major or school all together. Studies have indicated that when these individuals are isolated within a project team, they do not gain the benefits of working within a team because they are assigned the passive roles or choose to take on these roles [23]. Heller & Hollabaugh document the positive effects of heterogeneous groups, where students of the minority group feel more comfortable contributing [15].

In order to prevent this feeling of isolation, we have incorporated the idea of maintaining a balance of gender on a given team as a constraint to our team formation algorithm. Specific to engineering, women are typically the minority gender within a classroom setting. Instructors have the ability to specify whether or not they want to add this constraint.

2.1.3 Scheduling

Communication is an important factor in creating effective teams. When communication is limited, it becomes difficult to coordinate tasks among members, resolve conflicts, and develop interpersonal relations [12]. Studies have proven that both computer-mediated communication and face-to-face communication are an effective means of working together in a group [17]. However, it is shown that individuals prefer to interact in a face-to-face manner over virtual communication when working with others [24].

In order to enable the face-to-face interaction with other members of a team, it is important to place students in a group where there are common times to meet together. By asking students information regarding their schedule, we are able to incorporate this constraint when we provide suggestions on team formation.

Instructors have the ability to determine the minimum number of common time slots they want each team to have.

2.2 Existing Team Formation Tools

Through the research we have done, Catme is the only existing solution we have encountered that is utilized within classrooms that takes various constraints into consideration when forming teams [20]. Catme has a tool called Team-Maker that collects information from students, and utilizes the data gains to assign students to teams according to a criteria specified by the instructor [20]. Creating teams through a system like Catme has shown success within various classrooms [21]. More instructors are looking to algorithmic tools for team formation [31]. The Team-Maker tool utilized in creating teams demonstrated results of a more effective composition than teams created manually [28]. When comparing the results between teams created by the Team-Maker tool and teams created manually, teams created by the tool had a higher score average, but did no worse than those created by hand [8]. This means that the Team-Maker tool produced teams, where the overall score average of the teams was the same if not greater than the score average of the teams created by hand.

Our work will build on this current approach by allowing instructors, who are utilizing the tool, to have the power to move students around within the teams. While an algorithm can be utilized to form the initial teams and present suggested students to swap, instructors will be the ones to make the final decisions. In addition, we have improved upon the user interface to create a tool that is intuitive for our users to utilize.

2.3 Mixed Initiative

Mixed initiative is a flexible interaction strategy where our users work in conjunction with the automated system to contribute to the task [1]. Mixed initiative solutions have been utilized to solve several complex problems that involve scheduling, simulations, and management issues [7, 14].

Prior work has shown success in utilizing a mixed initiative interface. Cobi's scheduling tool is a mixed initiative interface that integrates preferences and constraints in order to build a schedule for a large conference [18]. Moreover, a mixed initiative solution was utilized to help complement a supervisor managing a team of robots [14]. Results have indicated that users prefer the mixed initiative support in a study, where mixed initiative was utilized to help customize an interface [6].

Mixed initiative interfaces have been utilized to solve complex problems, such as the aforementioned. Similar to the interface of Cobi, our proposed solution will allow professors to work in conjunction with Teammatic's algorithms by applying their own knowledge and expertise to create teams for their students. We will adopt the idea of offering suggestions for swaps to ease the process of swapping students from one team to another. In addition, we will integrate the idea an adaptable interface by allowing professors to easily move students from one team to another.

2.4 Prior Team Formation Strategies

Smartcrowd is an existing framework that focuses on improving collaborative crowdsourcing [29]. This framework takes into consideration various traits of the workers and the tasks assigned. Several different approaches were taken when implementing such an algorithm, including a greedy and an approximation approach. Our problem is similar to that of collaborative crowdsourcing in that we also want to find the most efficient way to group together individuals for the purpose of completing different tasks. However, these problems differ since crowdsourcing involves individuals who do not necessarily need to work together as a team, and can be completely unknown to one another [29].

Prior work has been conducted on forming teams online through social networks [3]. The algorithm proposed assigns various tasks for teams to deal with, while maintaining a fair distribution of work among the team and finding the minimum coordination cost. The work conducted by Anagnostopoulos et al. involves a static group of individuals who are connected within in a social network. Our set of individuals differs in that many may not know one another and their set of skills can vary drastically.

Diebel focuses on team formation for in-class group work [9]. She describes the educational benefits that students gain from learning from one another in a group setting. She stresses the importance of student interaction and how formation of the teams can have a major influence on the quality of collaborative learning. Our work differs in that the teams we form can be extended for a longer period of time, in addition to the shorter term teams, where students will need to meet together outside of the classroom [9].

Chapter 3

Needfinding: Current Team Formation Experiences

3.1 Introduction

An initial study was conducted in order to understand the experiences instructors have had with team formation. We asked 8 professors, lecturers, and teaching assistants to describe the importance of teams in their classes. All recruits were interviewed from the University of California, San Diego.

3.2 Method

Professors, lecturers, and teaching assistants were asked a series of questions in order to evaluate current procedures and tools utilized in forming teams. We targeted instructors who utilize teams within the courses they lead. The majority of courses were within the computer science field, but also consisted of cognitive science, business, and other engineering courses. Interviews typically lasted approximately 30-45 minutes long.

3.2.1 Initial Interview Questions

- What is the importance of teams in the classes you teach?
- How do you currently handle team formation?
- What are the current challenges in creating teams?
- What causes problems within teams?
  - How do you resolve these problems?
  - What can you do to prevent these problems?
- What traits of a team are the ones that are most successful?
- What do you think is important in team formation?
  - What are the factors that matter?
- How would you form teams if you had a good tool for it?

3.3 Findings

3.3.1 Importance of Teams Within a Given Course

Teams present a number of functions for students within the classroom setting. One of the main reasons instructors choose to utilize teams within their curriculum is to prepare students for their careers out in the real world. Teamwork helps students build upon the skills that would be necessary for working with others.

When we asked instructors what the importance of teams were we received the following responses.

"Teamwork is really important in software engineering and engineering in general. We want students to have that experience." - Software Engineering Professor

"Focused on teaching a student a very practical skill. This is how its done in the real world." - Cognitive Science Teaching Assistant

"To prepare students for the workforce. These days in almost every industry youre going to work as part of a team" - Business Lecturer

Allowing students to work within teams provides the opportunity for them to learn from one and another. Students will get the chance to ask each other questions and solve problems with their peers rather than directly going to the instructor for assistance. Teams enable students to engage in a collaborative learning environment. We asked instructors why they utilize teams in the various courses they teach.

"We want students to be learning from each other" - Software Engineering Professor

"A lot of learning happens in a much better way when people collaborate in teams. A team is an opportunity to transfer skills from from student to another" - Cognitive Science Professor

3.3.2 Current Methods for Team Formation

Instructors have different method of handling team formation. While some may manually create assigned teams for their students, most allow students to choose their own teams.

"Students pick the teams themselves, we arent forcing them" - Entrepreneurship Professor

The reason most instructors do not selectively place students together is because of the amount of time it would take to parse through the data and determine how to form teams.

"Instructor formed teams has the problem of taking up instructor time." - Cognitive Science Professor

Each course can range from having around 40 students to 200+ students, and given the amount of time professors have, it is not feasible to manually go through the information and group students together. If teams were assigned, most would randomly assign students together.

"Groups are formed randomly - Computer Science Teaching Assistant"

However, allowing students to create their own teams or randomly assigning teams may not produce the best results or provide the best learning experience for those in the class. Selectively placing students together, based off of data gathered, creates an improved team environment where students can develop necessary teamwork skills.

3.3.3 Challenges and Problems with Team Formation

Given their past experiences, it was found that instructors face similar challenges. Teams typically failed when there was not an even distribution of work for the given project and when members had conflicting goals. This boils down to the amount of commitment each member of the team wants to devote to the course. Instructors noticed that when members of the team have the same level of commitment to a course, the team does not encounter as many problems.

Challenges also occur when students do not interact well on an interpersonal level. The behavior and personality of each individual has an effect on the overall relationship between members of the team. The problem arises when students do not address the issue early on in the project and only bring up the conflict towards the end of the project. We received the following responses when asked what instructors believe make a successful team.

"The successful teams are a combination of having the same level of commitment to success in the class and being able to manage the personality complex." - Computer Science Professor

"The successful teams are the ones that have a shared goal and have a shared process." - Cognitive Science Professor

3.3.4 Desired Components of a Team Formation Tool

Flexibility was a key component that instructors look for in a team formation tool. Instructors would utilize a tool where they can easily gather data from their students and form teams given their own constraints. However, each constraint varied between each class and customization was key. Given the idea that instructors should form teams, many stated that they would consider this option only if shown the value of this process over their current process. Instructors would want to understand the reasoning behind why a given constraint is important within a team rather than blindly trusting a tool.

Chapter 4

Teammatic

The proposed solution is to create an interface that will make it easy for instructors to form teams in the classroom setting. This tool will be a mixed initiative interface where instructors will have the ability to work in conjunction with the Teammatic algorithm to form teams. Our user is the instructor and the system will be the interface used to form the teams. Having the user work with the system will allow a collaboration that will help to refine the results of the teams formed [16].

4.1 Design

Teammatic will allow instructors to import data about students within their class. The current version of Teammatic handles student availability, leadership attribute, gender, role distribution, and student preferences for possible teammates.

Once the data of student information is received, instructors will have the ability to choose constraints of how they want to create teams. The algorithm used for the team suggestions will account for these various constraints. After a suggested group of teams have been formed, the instructor will have the ability to modify these teams to his or her liking.

4.1.1 Student Roster

The student roster displays all the data gathered from the students of the course. Instructors will have the ability to sort the list of students in a variety of ways to easily find pertinent information. Figure 4.1 displays the interface of the student roster and the details of each student.

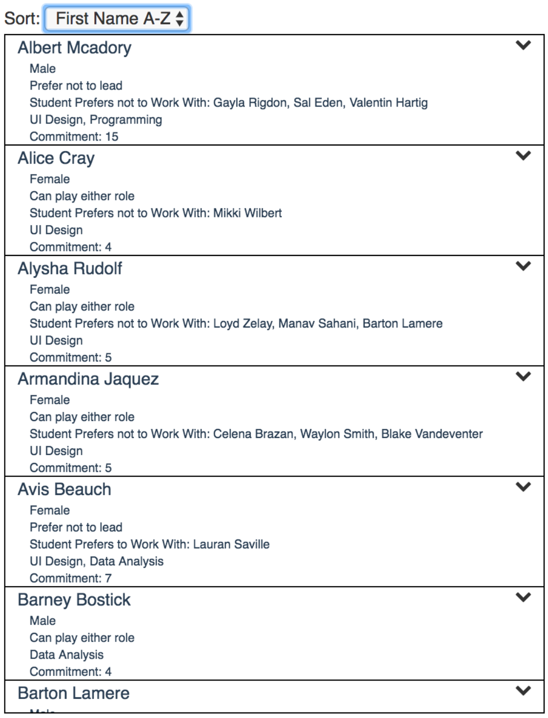

Figure 4.1: Student roster.

In addition, Teammatic provides a way to filter out certain data attributes for all students in the course. In Figure 4.2, we see the various filters that are available given the constraints that Teammatic currently accounts for.

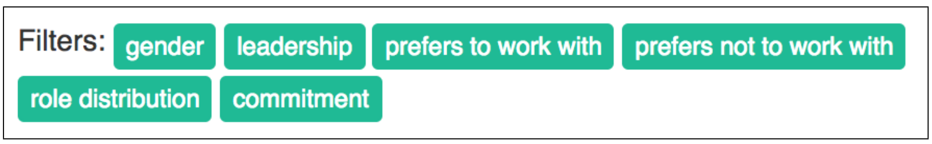

Figure 4.2: List of various filters.

Once a filter button is selected, the selected attribute will not be shown for all students displayed in the roster. Figure 4.3 displays an example of selecting certain attributes to hide in the roster.

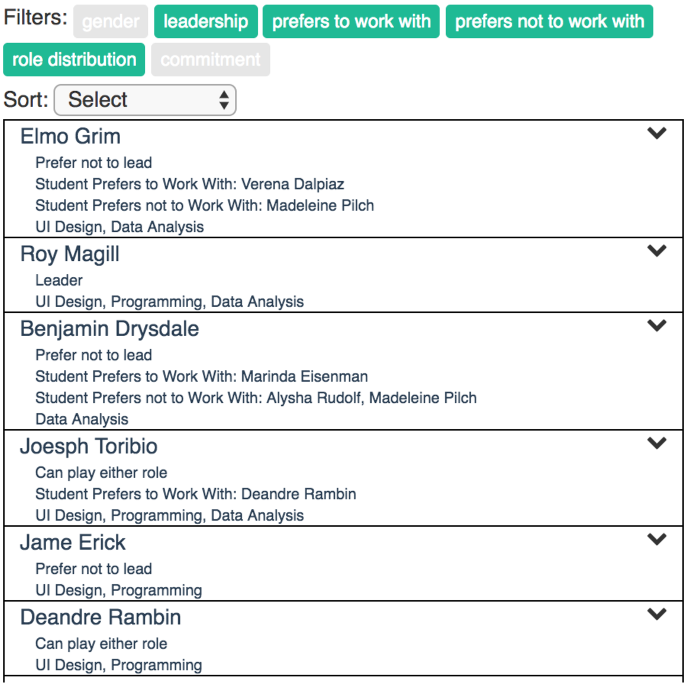

Figure 4.3: Example of selecting filters.

The student roster allows instructors to easily keep track of all the information they acquired from each student.

4.1.2 Constraints

Before deciding how to create teams, instructors will have the ability to input various constraints of how they would want the teams formed. In addition, they can place a weight on how much they would want to prioritize a certain constraint over the others. Figure 4.4 displays the constraint modal where instructors can input the constraints they desire.

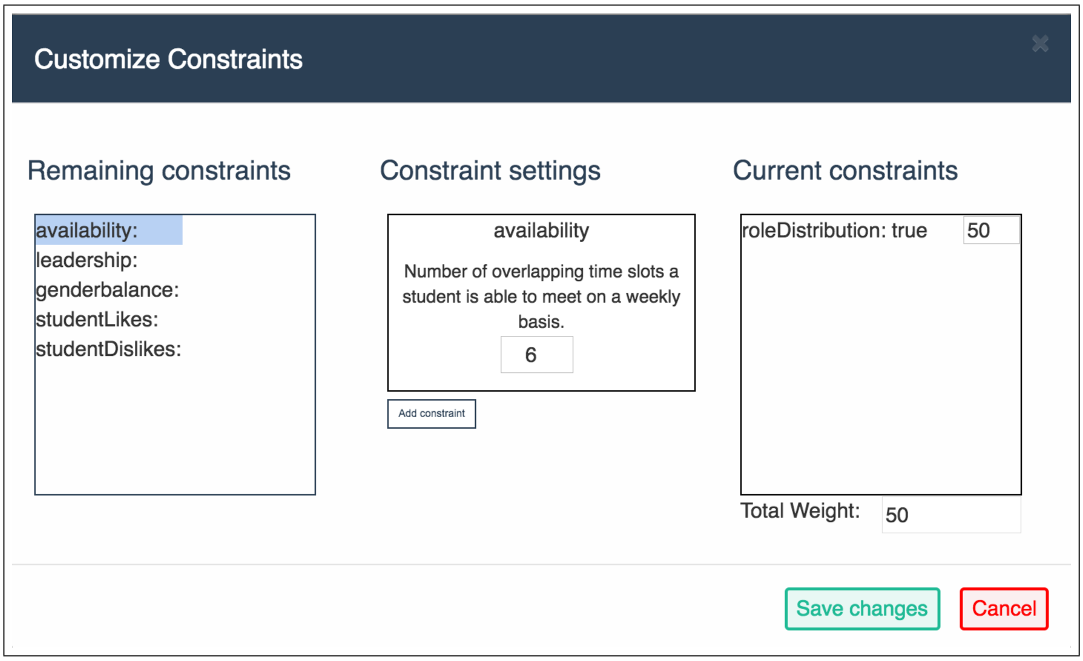

Figure 4.4: Interface where users can input their desired constraints.

4.1.3 Teams

Once the instructor has determined the various constraints they want to account for and inputted how many students they want in each team, they can select the "Create Teams" button to form a suggested group of teams. Figure 4.5 displays the output after all teams have been created. Teams are displayed in two columns, and Teammatic enables instructors to drag teams around as well, as collapse teams to only show the team name.

On the upper right hand corner of each team is a compatibility score. This compatibility score is determined by how well each team satisfies the given constraints. If a constraint is violated, a yellow warning sign will appear on the team

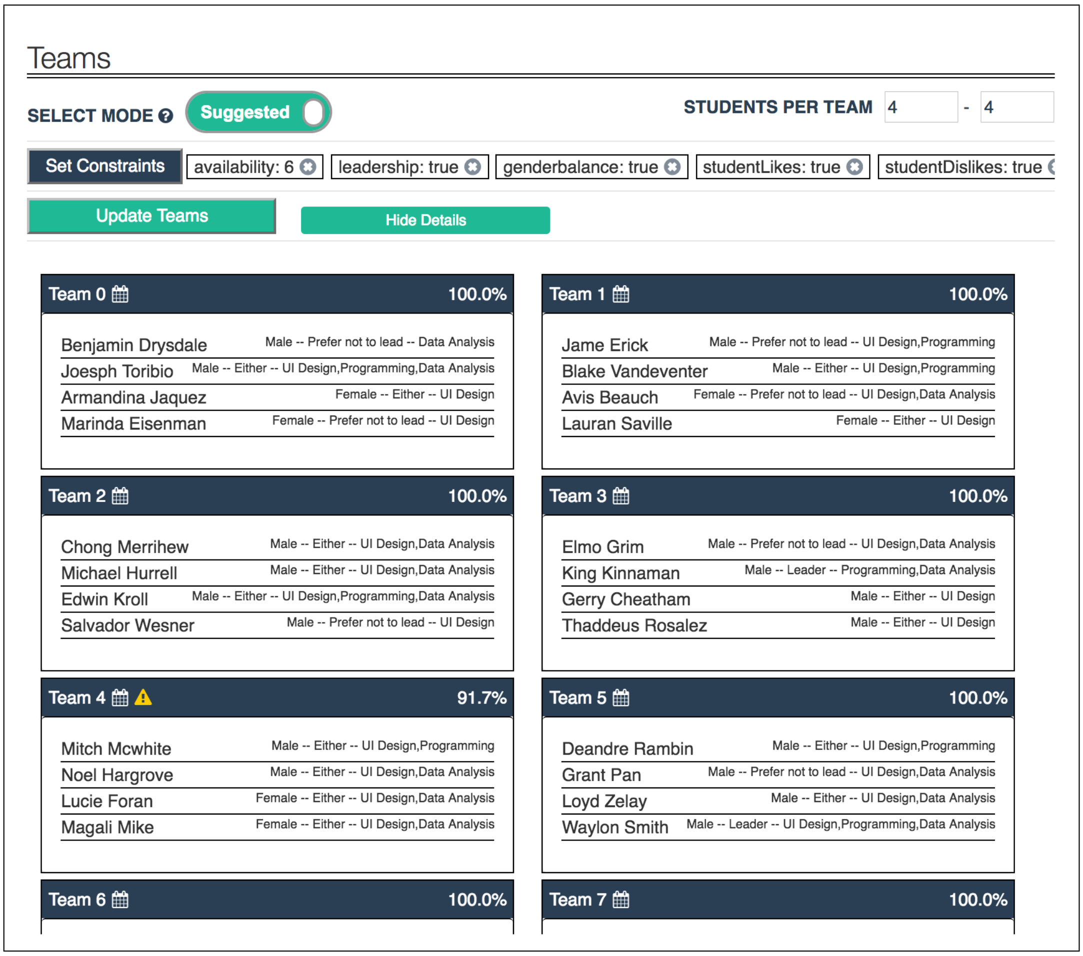

Figure 4.5: List of teams generated.

header. Figure 4.6 shows an example of a team that violates a constraint, and details about what constraint is violated.

In addition, each team contains a calendar that displays the overlapping time slots of all members on the team. By clicking the calendar icon on the team header, a calendar will appear to show where these time slots occur. Figure 4.7 displays what this calendar will look like. Each blue rectangle on the calendar represents a one-hour time slot where all members on the team are available.

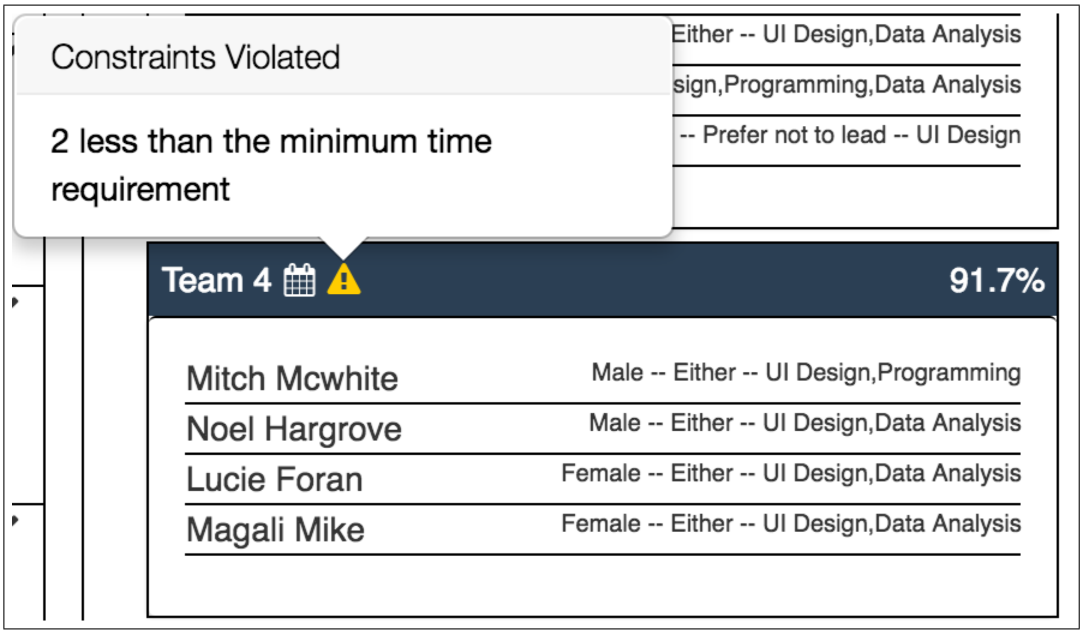

Figure 4.6: Displays the violated constraint for the team.

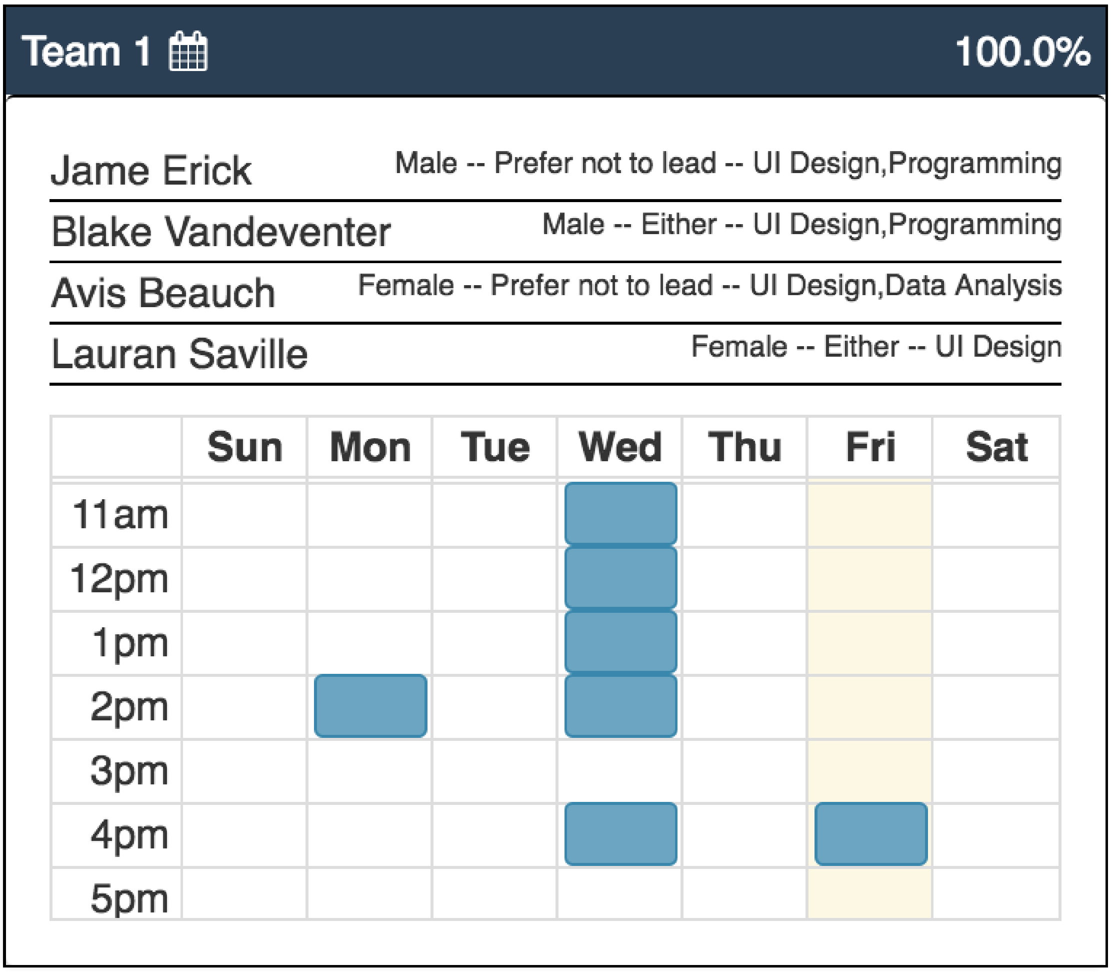

Figure 4.7: Common overlapping time slot for a given team.

4.1.4 Swapping Students

After the teams are formed, instructors will still have the ability to move students as they see fit. By clicking on an individual student, Teammatic will provide suggestions of whom to swap with, while maintaining the highest score average for both teams involved. Figure 4.8 displays an example of a suggested swap. The students highlighted in green are the ones Teammatic suggests to swap with the student highlighted in blue.

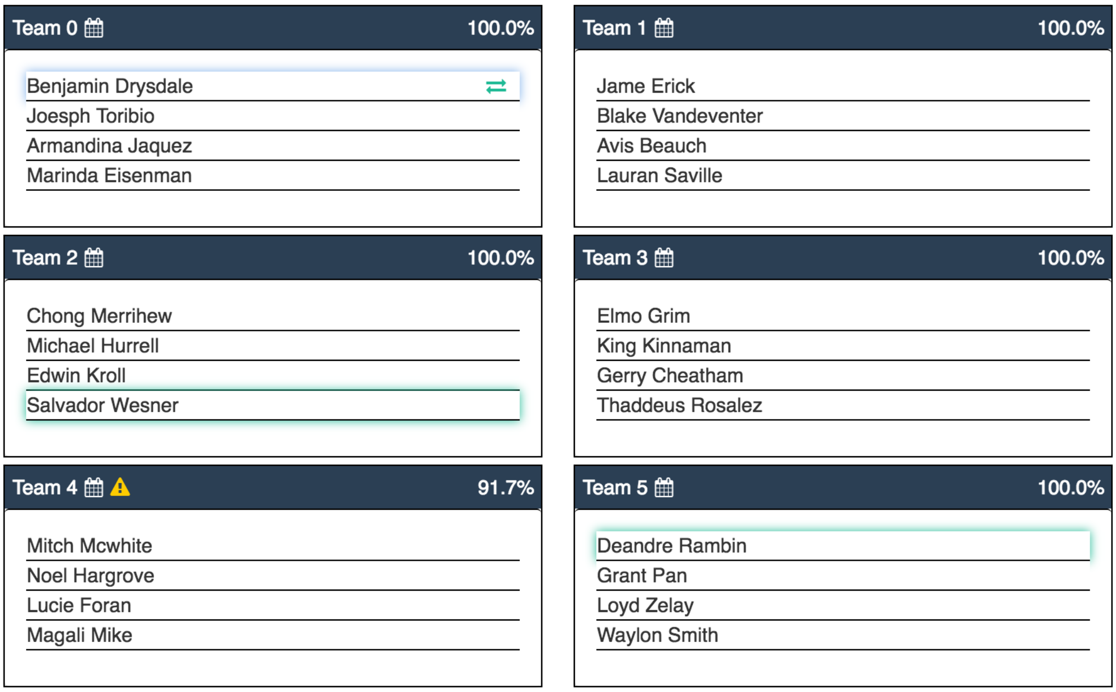

Figure 4.8: Example of a suggested swap.

Although suggestions can be provided for instructors, it is not required. Suggestions are made based off which student the instructor wants to swap, as well as the various attributes and constraints specified [section 4.3.5.2]. Instructors have the ability to simply drag and drop students from one team to another without the suggested swaps. This incorporates the idea of a mixed initiative interface, where instructors will work with the Teammatic system, but have the flexibility to make changes when it is desired.

4.2 System Overview

The system is built on a Meteor framework. Meteor is an open source platform for web, mobile, and desktop applications [32]. The user interface of Teammatic is built with HTML, JS, and CSS, while the algorithm is written in Python. We send the data from the front-end of the system to the server, which then calls the Python script in order to determine which students to place on each team. Once we run the algorithm and create a list of suggested teams with the data passed in, we send the data back to the server, which will output on the interface of Teammatic. Figure 4.9 displays the architecture of the system [32].

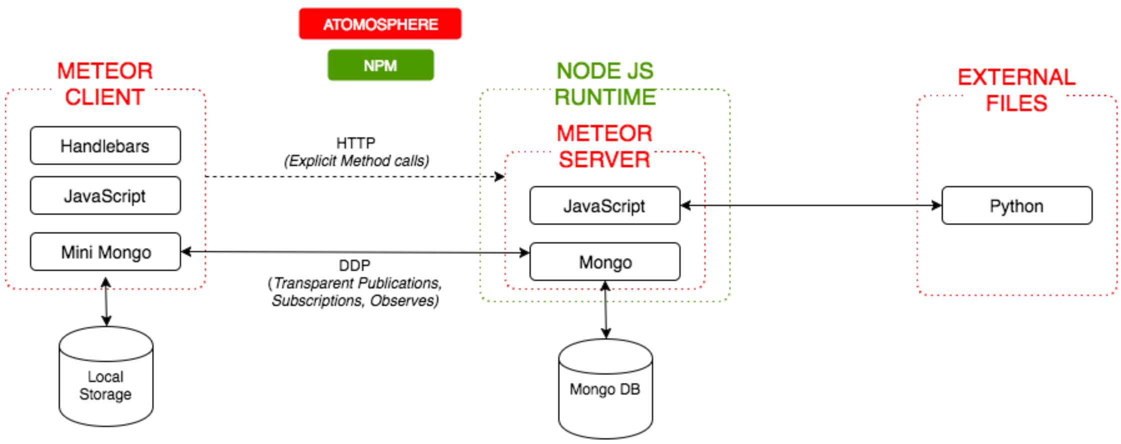

Figure 4.9: Meteor framework architecture.

4.3 Algorithm

4.3.1 Model

We began by defining a simple model for our algorithm on how to create the teams. We take as input, data that represents the students characteristics. Given the various data points obtained about each student, different constraints can be stated on how to create the teams. The objective of our algorithm is to create teams that maximize the overall compatibility score average.

4.3.2 Defining the Score Function

For each team, we will calculate its score based off the characteristics of the students that compose the team. As mentioned in Chapter 2, there are certain characteristics that contribute to creating an effective team. The team formation algorithm will select the teams that have the highest scores.

- Schedule_overlap : If the total number of overlapping slots is larger than the minimum, the score will be 1. If not, then the the score should return the total number of overlapping slots / minimum number of overlapping slots required

- Gender: 1 = female, 0 = male

If the total number of females on a team is 0 or 2 or more, we will give the team a score of 1 for the gender constraint. If the total number of females on the team is equal to 1, we will give the team a score of 0 for the gender constraint. The constraint is intended to ensure that the minority group is not isolated on a team [23].

- Leadership: 1 = leader, 0 = follower

If the total number of leaders on a team is 0 or 2 or more, we will give the team a score of 0 for the leadership constraint. If the total number of leaders on the team is equal to 1, we will give the team a score of 1 for the leadership constraint.

- Role_distribution: Number of roles satisfied in a team / total number of differing roles

Instructors will have the ability to determine the various roles that a student can play on a given team. For example, a team is composed of students who are only familiar with design and programming roles. If the instructor chooses to specify design, programming, and data analysis as the three differing roles, the team will receive a score of 2/3 for role distribution.

The score function will be the sum of the chosen characteristics with each characteristic multiplied by a weight (ex: w_1*schedule_overlap + w_2*gender + w_3*leadership + w_4*role_distribution). This weight, w_i, represents how much instructors will want to weight each characteristic when determining how to create teams. The score function is what will ultimately be used to evaluate the compatibility score of a team. Teammatic utilizes the compatibility score to determine which teams to select.

4.3.3 Constraints

We provide a way for instructors to input their own constraints, relative to their course. Ensuring that each student belongs to only one team and that each team has the correct number of students are the hard constraints that are satisfied for all teams created with Teammatic. In addition, if the instructor wants to honor student preferences, they can choose to make this an additional hard constraint.

All other constraints are flexible given the preferences of the instructor. Instructors will determine how important each constraint is when creating teams by assigning a weight to each constraint. These constraints will be incorporated into the score function to determine how compatible a team will be.

4.3.4 Objective

Our objective with this model is to maximize the score average across all teams, while minimizing the standard deviation. We want to obtain this maximum average while satisfying all given constraints. The given characteristics and constraints that we are to consider in the above example are subject to the instructor's discretion, and may vary depending on what the instructor is looking to have among teams in his or her class.

4.3.5 Implementation

4.3.5.1 Initial Team Formation

Once we have received the data of each individual student, we will specify which constraints to take into consideration. These constraints can include the minimum number of overlapping times within a team's combined schedule, balancing out gender within each team, ensuring that one leader exists within each group, distributing the roles across teams, or allowing students to specify preferences for teammates. Each of the teams are given a score based off of the importance assigned to each constraint and how well they satisfy the requirements. The algorithm will compute teams on the objective of maximizing the overall score average within the entire set of teams.

To select the teams, we utilize a greedy approach. We first generate all possible combinations of teams with the given team size. Next, we filter these combinations with the hard constraints. Currently, student preferences is the only hard constraint accounted for. We take into consideration all the students and the given constraints as inputted by the instructor. The students who are most constrained will be placed on a team first. By most constrained, we mean the students who placed in the least number of teams when generating all possible combinations. We select the highest scoring team with the student who is most constrained. Once the team is selected, we eliminate all students who were placed on the selected team. From there, we repeat the process of selecting the team for the next most constrained students with the highest possible score, until all students have been placed on a team.

Given the idea of incorporating constraints, it is possible that some students are left remaining without a team. With the remaining students, we loop through the already selected teams and place these students on a team that generates the highest score, while still maintaining the list of constraints.

4.3.5.2 Swapping Students

Once the teams have been formed through the algorithm, if instructors want to swap students from one team to another, Teammatic provides suggested swaps of students. However, instructors are not forced to swap students only based off of the suggest swaps; they are the ones who make the final decision.

The swapping algorithm takes into consideration which student the instructor chooses to swap, their current teams, and the current constraint list. It will then provide suggestions with other individuals in the course which maximizes the compatibility scores of both the teams if the swap were to occur.

4.3.5.3 Constraint Violation

Given the dataset of student information and the list of constraints chosen by the instructor, there are times where constraints may be violated when creating the teams. Once the teams are created, we check to see which constraints are violated and provide the feedback of each constraint violation for the corresponding team on the interface.

4.3.6 Evaluation of the Results

To evaluate the effectiveness and efficiency, we ran the algorithm on several different constraints with varying team sizes. We measured the amount of time it takes for Teammatic's algorithm to output results and the overall compatibility score average of the resulting teams.

Team Size		All Constraints	Only Soft Constraints	Only Hard Constraints	No Constraints
2-3 Students					
	Time (in seconds)	4.572	4.63	4.633	3.946
	Team Score Avg.	96.80%	97.50%	N/A	N/A
3-4 Students					
	Time (in seconds)	52.019	64.95	49.593	52.118
	Team Score Avg.	97.10%	98.30%	N/A	N/A
4-5 Students					
	Time (in seconds)	419.274	807.529	407.157	608.371
	Team Score Avg.	96.20%	0.962	N/A	N/A

Figure 4.10: The performance of the algorithm given different constraints.

Figure 4.10 displays the various performance metrics that result from the algorithm described above. Several different constraints were inputted to create teams from a data set of 52 students. Our algorithm shows the variance in times to compute teams based off how many students the instructor specifies to place on each team. The larger the groups, the longer it takes to create a list of suggested teams. The reason behind this is because our algorithm starts by computing all possible combinations of students before determining which to place on a given team.

The average scores of the teams created are not affected by the team size. The algorithm utilized selects teams that maximize the overall average score.

Chapter 5

Participants Forming Teams

5.1 Introduction

A controlled study was conducted in order to gain insight behind manual team formation processes compared to Teammatic. 14 teaching assistants from various courses were recruited to participate in the study and were contacted via email. All teaching assistants involved currently or have previously taught a team-based course.

During the study, we observed how teaching assistants interacted with Teammatic as well as methods they utilized in order to satisfy the requirements provided to them to form teams.

5.2 Method

5.2.1 Instructions

The following instructions were provided to the teaching assistants that participated in the study:

"Pretend that you are the teaching assistant for a project based course. The professor of the course wants you to create teams with the following requirements. These are hard constraints when creating teams:

- 4 students on each team
- Ensure that each student is on one team
- Student preferences are honored
  - Students will not be placed on a team with those they list under the Prefers not to work with section
  - Students will only be guaranteed on a team with those they list under Prefers to work with if both students involved mutually request one another

The instructor also wants you to create teams with the following soft constraints in mind to the best of your ability. Each of these constraints are equally as important to this instructor:

- Teams with distributed roles (ex: a team has a programmer, designer, and analyst rather than all programmers)
- At least 1 leader on each team
- Either 0 females or 2 or more females on each team
- At least 6 overlapping time slots in common"

5.2.2 Process

Instructors began the study by reading the instructions provided above. They were given time to manually create the teams with the Teammatic interface. The dataset that was utilized in this study derives from real student data of a course. Once the instructors felt that they were satisfied with teams they had formed, questions were asked regarding what they found difficult, their thought process of creating the teams, and their confidence level in the manual teams they created.

Next, the participants were shown the automated teams created by Teammatic. We asked how they felt these teams compared to their own and how confident they were in the teams formed automatically. Participants were then given a scenario that described two students who mutually preferred to work together, but as the instructor, they have prior knowledge about the students and know that when the two students are placed together, they tend to act dishonestly. It was then observed what the participant would do in this situation, and how they would change the automated teams.

We concluded the study by asking the participant to evaluate their experience and results with the manual team formation process compared to that of the automated version in a short interview.

5.3 Results

It was hypothesized that the scores of the manually created teams would be at most equivalent or less than the compatibility scores of the teams generated from the automatically created teams. In addition, we hypothesized that with Teammatic, the amount of time it would take to create these teams would be significantly improved. Both of these hypotheses were proven true.

When given the same constraints as the participants of the study, Teammatic produced teams with an overall average compatibility score of 98.077%. The average compatibility score for the manually created teams was 84.2775%. Given the same constraints with different data, Teammatic produced an average compatibility score of 91.450%. Figure 5.1 displays the results. There was a standard error of 1.246% for the manually created teams and 0.46% for Teammatic with varying data. The data gathered also shows that no individual was able to produce an overall average score between the teams with as high of a percentage.

This task took participants an average of 30 minutes 50 seconds to complete. With the automated tool, forming teams with the same requirements takes an average of 44.673 seconds. Given the same constraints with different data, Team-

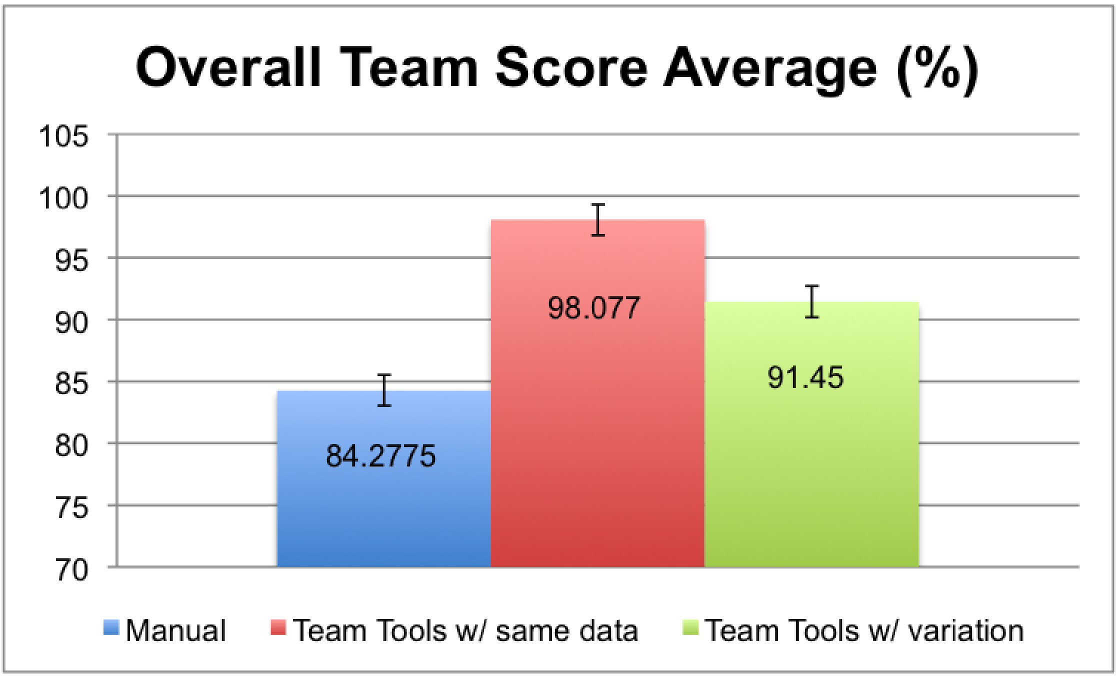

Figure 5.1: Overall Team Compatibility Score Average.

matic takes an average of 43.688 seconds to form teams. Figure 5.2 displays the results. When asked what the participant found difficult in creating teams, the most common response was that it was hard to keep track of all the information. Given the many constraints listed in the instructions, many chose to ignore the soft constraints and focused their time on the hard constraints listed. After a certain point in the process of creating teams, it was a common trend that participants would start to randomly place students together without any regard to the constraints given. Teammatic eases the process of team formation and provides a solution to handling several different constraints and data when creating teams.

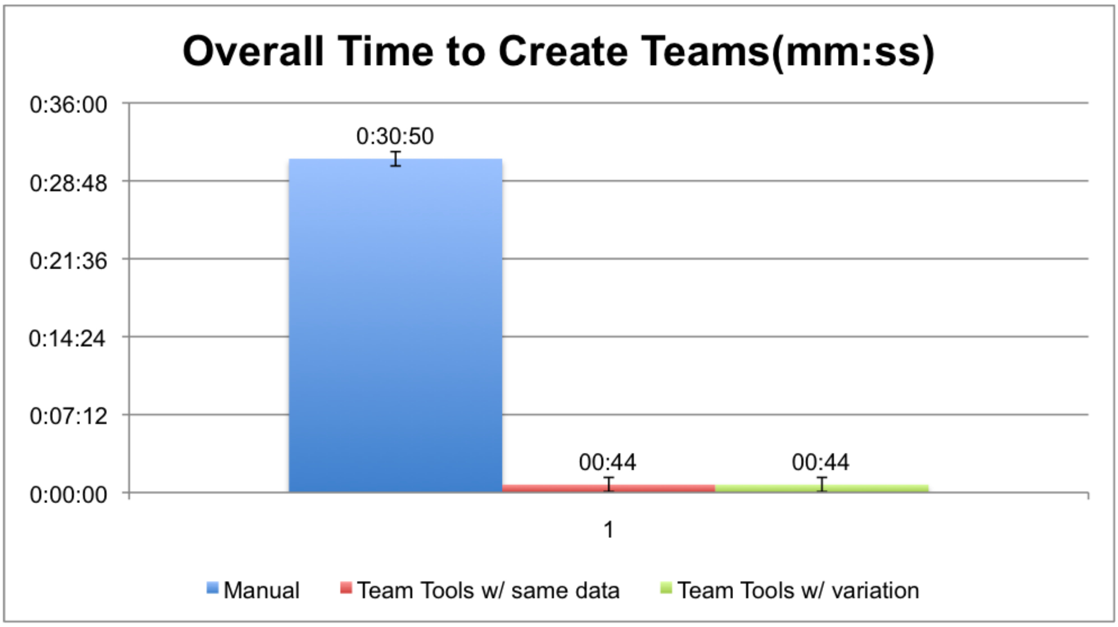

Figure 5.2: Average amount of time taken to form teams.

Participants were given a scenario with two students in the dataset as explained above. Since mutual preferences place students on the same team, Teammatic automatically accounts for this constraint. Participants were asked what they would do to handle this situation. Results show that all participants would want to separate the two students and prevent them from being on the same team. To do so, many would randomly swap students or they would look for students with similar attributes, but had concerns with maintaining the several other constraints in the requirements. To handle this situation, Teammatic provided suggestions of who to swap students with, keeping in mind the constraints listed.

Teammatic focuses on improving the experience of the instructor when forming teams. By measuring the confidence participants had in the teams formed, we were able to evaluate the qualitative difference between the manually formed teams and the teams formed automatically through Teammatic. When asked about the confidence of the manually formed teams on a scale of 1 being least confident and 7 being most confident, participants rated themselves with an average score of 4.333, whereas they rated the teams formed by Teammatic with an average score of 6.208. Figure 5.3 displays the results. All participants rated the confidence of the teams formed by Teammatic either equal to the teams formed manually or greater. The reasoning many had was that Teammatic was able to successfully handle all constraints given in the requirements when the participants could not.

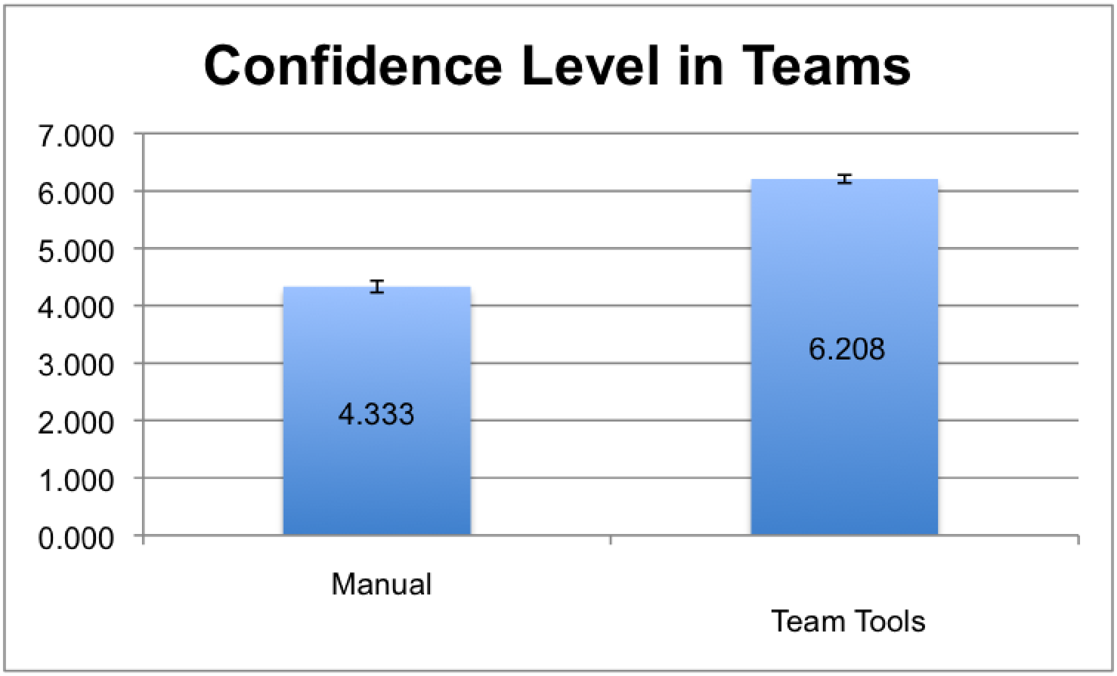

Figure 5.3: The confidence participants had in the teams formed.

Chapter 6

Case Study: Instructor Utilizing Teammatic

6.1 Introduction

An initial case study was conducted with a prototype of Teammatic. The professor who participated in the study teaches a cognitive science course that explores the intersection of social behavior and computational systems.

According to the professor, teams are an important and essential component of the curriculum for two reasons. The first is the practical reason because with so many students in his course, grading is more manageable with teams. The second is a pedagogical reason, since these students will often be working with teams out in the real world, and it is important to learn how to work with others, while also learning how to be a good teammate themselves. In addition, having teams allows students to take on more complex projects. Specific to this course, students work on a project that has a design, programming, and analysis component, so having teams with students skilled in these areas will help.

Typically when creating teams, the professor looks for certain characteristics of individuals to pair together. His manual method for creating teams starts off by finding the students who are most interested in the content and seem to be good at leading others. With these students, he spreads them out across the teams. Around this, he strives for diversity in terms of skills and demographics. If possible, he will then account for schedule to try and get as much overlapping time between students on a team. Without a tool to account for all these different data points, this process would typically take hours for the professor to complete.

6.2 Method

The professor sent out a survey to gather data from the students of their course. Once the students had filled out the survey, we generated the CSV file for the instructor to input into Teammatic. Some manual data manipulation was required to ensure the data could be read with Teammatic. At this point, we instructed the professor to utilize Teammatic to create teams. As he was utilizing the tool, we utilized the think aloud protocol where he would describe his thought process when navigating through the tool. By utilizing the think aloud protocol, we are able to understand when the instructor may be confused, and what he is thinking about as he is in the process of creating the teams. This information was utilized to further improve both the design and algorithm of Teammatic.

6.2.1 Constraints

Several constraints were important to the instructor when creating teams. The course had 52 students enrolled and the professor wanted teams of 4. In addition, the professor wanted each team to have at least 6 overlapping time slots where all members are available, at least one leader, either two or more females per team or none at all, and individuals with a variety of roles they could contribute to the team. The instructor chose to give each of these constraints equal weight when inputted into Teammatic – this meant each constraint was given a weight of 25% out of 100%. The constraints involving schedule, gender, leadership, and roles were soft constraints for teams, where the algorithm would strive to get the best scores with the data inputted.

The professor also wanted student preferences to be acknowledged. These were hard constraints for our algorithm. If a mutual preference was stated for students wanting to be on a team together, or if a student did not want to be placed on a team with another student, the teams should be created accordingly.

6.2.2 Survey

An initial survey was sent to the students of the course three days prior to conducting the study. Students filled out a Google Form with the following questions:

- What is your full name?
- What is your student id?

- What is your gender? Students chose from the following options[1]:
  - Male
  - Female
- How would you describe your leadership quality? Students chose from the following options:
  - I prefer to be a leader on the team
  - Prefer if someone else leads the team
  - I can play either role
- What is your availability?
  - Students were given a Doodle link to fill out their weekly availability for the entirety of the course. The time slots ranged from Sunday to Saturday, 8am to 8pm with 1 hour time increments.
- Approximately how many hours per week do you intend to commit to this team project?
  - Students were asked to enter a whole number to approximate the number of hours per week they hope to commit to the course
- Select all roles you feel you can contribute on a team. Students chose from the following options:

[1] Teammatic will change this survey question to be more gender inclusive in the future. For this initial case study, our algorithm does not account for other gender identities.

  - UI Design
  - Programming
  - Data Analysis

- If desired, list one student you would prefer to work with.
  - Students were given the option to request one student they would like to work with. This preference would only be considered if there was a mutual request between the two students.
- Students you prefer not to work with
  - Students were allowed to list as many people that they desired whom they did not want to be placed on a team with.
- Any other preferences or constraints that you want the instructors to know?

6.3 Results

Teammatic helped in dealing with several constraints that the professor wanted within a team. Teammatic created teams that accounted for overlapping schedule, gender, leadership, and role distribution. The algorithm took approximately 1 minute to compute teams with those constraints.

However, the tool failed in accounting for student preferences. Students had the option to determine who they wanted to work with and did not want to work with, but Teammatic was not yet able to handle this constraint. In order to account for the student preferences, the professor had to manually swap students around. The process of manual swapping was difficult because the instructor had to account for all the hard constraints, while still trying to create teams that satisfied the soft constraints. By incorporating the idea of a mixed initiative, Teammatic could be improved by displaying when constraints are broken after swapping students, and providing suggested swaps for instructors when creating teams.

Overall, the instructor felt pretty confident in the resulting teams produced by Teammatic. When manually swapping students, he ended up just looking at the schedule overlap and whether or not any hard constraints were violated. At times, swapping students from one team to another caused other constraint violations, which made it difficult. Although manual swapping was involved to satisfy the hard constraints, the professor does believe that these teams created are better than those that would have been created if chosen randomly. According to the professor, Teammatic is important because when creating teams he would not have to guess on how to place students together.

Chapter 7

Conclusion and Future Work

Teammatic provides an alternative improved approach to creating teams within the classroom setting. We have demonstrated the benefit of strategically placing students together, and how current manual processes make it difficult to achieve results quickly. With the mixed initiative interface, the instructors forming the teams have the power to move students around and to work in conjunction with the algorithm Teammatic provides. Experimental results have demonstrated the advantages that Teammatic provides over manual processes and have shown positive results when comparing compatibility scores, time, and confidence.

7.1 Further Interface Improvements

In future work, we plan to incorporate further interface improvements, as well as add more features to the system. Currently, Teammatic does not have a fully integrated process. It would be ideal to incorporate the entire team formation process into Teammatic, from getting data from students to forming teams to peer feedback at the end of the course. We plan to have a fully integrated system before releasing Teammatic to allow others to use.

Furthermore, we seek to add additional features to improve the experience of creating teams. Currently given that Teammatic utilizes a greedy algorithm, it is capable of generating only a single option for instructors given a data set with certain constraints. By switching from a greedy approach to an optimization approach, we can provide multiple options of team suggestions for those utilizing our tool. In addition, as we have previously evaluated, the speed of the algorithm can be further improved for larger groups of students.

It would also be helpful to allow instructors to lock in certain teams they want to keep and rerun the algorithm to create new teams with the remaining students. In addition being able to add customized constraints will allow for better flexibility for instructors to create the teams they desire. By pursuing these changes, Teammatic will provide an even better approach to the team formation process.

7.2 Future Studies

We have shown the benefit of utilizing Teammatic to create teams over manual methods. To further validate the benefit of Teammatic, additional studies would be useful. A more long term study is required to understand the performance of teams created with Teammatic over old practices. We hypothesize that by utilizing Teammatic, teams will perform better than the teams generated through old practices. Teammatic has the potential to change the way teams are created in the classroom setting.

7.3 Discussion

Teammatic has the potential to change the way team formation is implemented not only within education, but also in the workforce. By strategically placing individuals together based off of their unique characteristics, we can help to further develop teamwork skills. Beyond education, Teammatic may be used in a capacity to create effective teams in industry, in turn yielding better team performance. Users will have the ability to determine what they believe is important in a team, and Teammatic will automatically create teams with the given characteristics. Therefore, Teammatic will revolutionize the process of team formation by reducing the amount of time it takes to create teams and optimizing the user's team preferences.

Bibliography

[1] Allen, J.E., C.I. Guinn, and E. Horvtz. "Mixed-initiative Interaction." Mixed-initiative Interaction. IEEE, 06 Aug. 2002. Web. 02 Nov. 2016.

[2] Ammeter, Anthony P. and Dukerich, Janet M. (2002) Leadership, Team Building, and Team Member Characteristics in High Performance Project Teams, Engineering Management Journal, 14:4, 3-10, DOI: 10.1080/10429247.2002.11415178

[3] Anagnostopoulos, Aris, Luca Becchetti, Carlos Castillo, Aristides Gionis, and Stefano Leonardi. "Online Team Formation in Social Networks." Proceedings of the 21st International Conference on World Wide Web - WWW '12 (2012): n. pag. Web.

[4] Avolio, B. J., Jung, D. I., Murry, W., & Sivasubramaniam, N. (1996). Building highly developed teams: Focusing on shared leadership processes, efficacy, trust and performance. In M.

[5] Bell, Suzanne T. "Deep-level Composition Variables as Predictors of Team Performance: A Meta-analysis." Journal of Applied Psychology 92.3 (2007): 595-615. Web.

[6] Bunt, Andrea, Cristina Conati, and Joanna Mcgrenere. "Supporting Interface Customization Using a Mixed-initiative Approach." Proceedings of the 12th International Conference on Intelligent User Interfaces - IUI '07 (2007): n. pag. Web.

[7] Butler, Keith A., Jiajie Zhang, Chris Esposito, Ali Bahrami, Ron Hebron, and David Kieras. "Work-centered Design." Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '07 (2007): n. pag. Web.

[8] Cavanaugh, R., Ellis, M., Layton, R., & Ardis, M. (2004). Automating the Process of Assigning Students to Cooperative Learning Teams. Proceedings of the 2004 ASEE Annual Conference , CD-ROM.

[9] Deibel, Katherine. "Team Formation Methods for Increasing Interaction during In-class Group Work." ACM SIGCSE Bulletin 37.3 (2005): 291. Web.

[10] Dreu, Carsten K. W. De, and Laurie R. Weingart. "Task versus Relationship Conflict, Team Performance, and Team Member Satisfaction: A Meta-analysis." Journal of Applied Psychology 88.4 (2003): 741-49. Web.

[11] Evans, Charles R., and Kenneth L. Dion. "Sign In: Registered Users." Group Cohesion and Performance - Aug 18, 2016. Sage Publications, Inc., 1991. Web. 24 Feb. 2017.

[12] Fransen, Jos, Paul A. Kirschner, and Gijsbert Erkens. "Mediating Team Effectiveness in the Context of Collaborative Learning: The Importance of Team and Task Awareness." Computers in Human Behavior 27.3 (2011): 1103-113. Web.

[13] Hansen, Randall S. (2006) Benefits and Problems With Student Teams: Suggestions for Improving Team Projects, Journal of Education for Business, 82:1, 11-19, DOI:10.3200/JOEB.82.1.11-19

[14] Hardin, Benjamin, and Michael A. Goodrich. "On Using Mixed-initiative Control." Proceedings of the 4th ACM/IEEE International Conference on Human Robot Interaction - HRI '09 (2009): n. pag. Web.

[15] Heller, P., & Hollabaugh, M. (1992). Teaching problem solving through cooperative grouping. Part 2: Designing problems and structuring groups. Am. J. Phys., 60(7), 637-644.

[16] Horvitz, Eric. "Principles of Mixed-initiative User Interfaces." Principles of Mixed-initiative User Interfaces. CHI'99, 1999. Web. 02 Nov. 2016.

[17] Jonassen, David H., and Hyug Kwon. "Communication Patterns in Computer Mediated versus Face-to-face Group Problem Solving." Educational Technology Research and Development 49.1 (2001): 35-51. Web.

[18] Kim, Juho, Haoqi Zhang, Paul Andr, Lydia B. Chilton, Wendy Mackay, Michel Beaudouin-Lafon, Robert C. Miller, and Steven P. Dow. "Cobi." Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology - UIST '13 (2013): n. pag. Web.

[19] Lepine, Jeffery A., Ronald F. Piccolo, Christine L. Jackson, John E. Mathieu, and Jessica R. Saul. "A Meta-Analysis Of Teamwork Processes: Tests Of A Multidimensional Model And Relationships With Team Effectiveness Criteria." Personnel Psychology 61.2 (2008): 273-307. Web.

[20] Loughry, M. L., M. W. Ohland, and D. J. Woehr. "Assessing Teamwork Skills for Assurance of Learning Using CATME Team Tools." Academy of Management Proceedings 2013.1 (2013): 14820. Web.

[21] Maneeratana, Kuntinee, and Angkee Sripakagorn. "Use of CATME for Teamwork Assessment in Engineering Projects." Academia.edu - Share Research. N.p., n.d. Web. 29 Dec. 2016.

[22] Mehra, Ajay, Brett R. Smith, Andrea L. Dixon, and Bruce Robertson. "Distributed Leadership in Teams: The Network of Leadership Perceptions and Team Performance." The Leadership Quarterly 17.3 (2006): 232-45. Web.

[23] Oakley, Barbara, Rebecca Brent, Richard M. Felder, and Imad Elhajj. "Turning Student Groups into Effective Teams." New Forums Press, Inc. (2004): n. pag. Web.

[24] Ocker, Rosalie J., and Gayle J. Yaverbaum. "Asynchronous Computer-mediated Communication versus Face-to-face Collaboration: Results on Student Learning, Quality and Satisfaction." Group Decision and Negotiation 8.5 (1999): 427-40. Web.

[25] Pearce, C. L., & Sims Jr., H. P. (2002). Vertical versus shared leadership as predictors of the effectiveness of change management teams: An examination of aversive, directive, transactional, transformational and empowering leader behaviors. Group Dynamics, 6, 172197.

[26] Plowman, N. "Seven Factors of Successful Teams: The Keys to Ensure High Team Performance." Brighthub Project Management. N.p., 15 June 2015. Web. 02 Nov. 2016.

[27] Richter, A. W., J. F. Dawson, and M. A. West. "The Effectiveness of Teams in Organizations: A Meta-analysis." The International Journal of Human Resource Management 22.13 (2011): 2749-769. Web.

[28] Roberts, Matthew W. "Using CATME to Create Student Teams and to Evaluate Team-Member Effectiveness." ResearchGate. N.p., n.d. Web. 29 Dec. 2016.

[29] Roy, Senjuti Basu, Ioanna Lykourentzou, Saravanan Thirumuruganathan, Sihem Amer-Yahia, and Gautam Das. "Knowledge-intensive Organizations." The New Knowledge Workers (n.d.): n. pag. 7 Jan. 2014. Web.

[30] Sarin, Shikhar, and Christopher Mcdermott. "The Effect of Team Leader Characteristics on Learning, Knowledge Application, and Performance of Cross-Functional New Product Development Teams." Decision Sciences 34.4 (2003): 707-39. Web.

[31] Jahanbakhsh, Farnaz, Wai-Tat Fu, Karrie Karahalios, Darko Marinov, and Brian Bailey. "You Want Me to Work with Who? Stakeholder Perceptions of Automated Team Formation in Project-based Courses." CHI (2017): n. pag. Web.

[32] "Meteor: Build IOS and Android Apps That Are a Delight to Use." MongoDB. N.p., 27 Apr. 2015. Web.